

8月22日

笔记

跨域问题:

出于浏览器的同源策略限制。同源策略 (Sameoriginpolicy) 是一种约定, 它是浏览器最核心也最基本的安全功能, 如果缺少了同源策略, 则浏览器的正常功能可能都会受到影响。可以说Web是构建在同源策略基础之上的, 浏览器只是针对同源策略的一种实现。同源策略会阻止一个域的 javascript 脚本和另外一个域的内容进行交互。所谓同源 (即指在同一个域) 就是两个页面具有相同的协议 (protocol) , 主机 (host) 和端口号 (port)

springboot跨域配置

```
@Configuration
public class CorsConfig {
    private CorsConfiguration buildConfig() {
        CorsConfiguration corsConfig= new CorsConfiguration();
        corsConfig.addAllowedOrigin("*");
        corsConfig.addAllowedHeader("*");

        corsConfig.addAllowedMethod("OPTIONS");
        corsConfig.addAllowedMethod("HEAD");
        corsConfig.addAllowedMethod("GET");
        corsConfig.addAllowedMethod("PUT");
        corsConfig.addAllowedMethod("POST");
        corsConfig.addAllowedMethod("DELETE");
        corsConfig.addAllowedMethod("PATCH");
        //同上 所有方法
        corsConfig.addAllowedMethod("*");
        return corsConfig;
    }

    @Bean
    public CorsFilter corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        // 配置所有请求
        source.registerCorsConfiguration("/**", buildConfig());
        return new CorsFilter(source);
    }
}
```